Notebook

June 20, 2025

[ ]:
```python
# Import Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import (accuracy_score, confusion_matrix,␣
 ↪classification_report,
                             roc_auc_score, roc_curve, precision_recall_curve)
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

[ ]:
```python
#  Load and Inspect Data

df = pd.read_csv("student_depression_dataset.csv")

print(df.head())
print(df.info())
print(df.describe(include='all'))
print("\nMissing values per column:\n", df.isnull().sum())
```

```
   id  Gender   Age           City Profession  Academic Pressure  \
0   2    Male  33.0  Visakhapatnam    Student                5.0
1   8  Female  24.0      Bangalore    Student                2.0
2  26    Male  31.0       Srinagar    Student                3.0
3  30  Female  28.0       Varanasi    Student                3.0
4  32  Female  25.0         Jaipur    Student                4.0

   Work Pressure  CGPA  Study Satisfaction  Job Satisfaction  \
0            0.0  8.97                 2.0               0.0
1            0.0  5.90                 5.0               0.0
2            0.0  7.03                 5.0               0.0
3            0.0  5.59                 2.0               0.0
4            0.0  8.13                 3.0               0.0
```

```
       Sleep Duration Dietary Habits   Degree  \
0          '5-6 hours'        Healthy  B.Pharm
1          '5-6 hours'       Moderate      BSc
2  'Less than 5 hours'        Healthy       BA
3          '7-8 hours'       Moderate      BCA
4          '5-6 hours'       Moderate   M.Tech

  Have you ever had suicidal thoughts ?  Work/Study Hours Financial Stress  \
0                                   Yes               3.0              1.0
1                                    No               3.0              2.0
2                                    No               9.0              1.0
3                                   Yes               4.0              5.0
4                                   Yes               1.0              1.0

  Family History of Mental Illness  Depression
0                               No           1
1                              Yes           0
2                              Yes           0
3                              Yes           1
4                               No           0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27901 entries, 0 to 27900
Data columns (total 18 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     27901 non-null  int64
 1   Gender                                 27901 non-null  object
 2   Age                                    27901 non-null  float64
 3   City                                   27901 non-null  object
 4   Profession                             27901 non-null  object
 5   Academic Pressure                      27901 non-null  float64
 6   Work Pressure                          27901 non-null  float64
 7   CGPA                                   27901 non-null  float64
 8   Study Satisfaction                     27901 non-null  float64
 9   Job Satisfaction                       27901 non-null  float64
 10  Sleep Duration                         27901 non-null  object
 11  Dietary Habits                         27901 non-null  object
 12  Degree                                 27901 non-null  object
 13  Have you ever had suicidal thoughts ?  27901 non-null  object
 14  Work/Study Hours                       27901 non-null  float64
 15  Financial Stress                       27901 non-null  object
 16  Family History of Mental Illness       27901 non-null  object
 17  Depression                             27901 non-null  int64
dtypes: float64(7), int64(2), object(9)
memory usage: 3.8+ MB
None
                  id Gender           Age    City Profession  \
count   27901.000000  27901  27901.000000   27901      27901
```

```
unique           NaN      2           NaN      52         14
top              NaN   Male           NaN  Kalyan    Student
freq             NaN  15547           NaN    1570      27870
mean    70442.149421    NaN     25.822300     NaN        NaN
std     40641.175216    NaN      4.905687     NaN        NaN
min         2.000000    NaN     18.000000     NaN        NaN
25%     35039.000000    NaN     21.000000     NaN        NaN
50%     70684.000000    NaN     25.000000     NaN        NaN
75%    105818.000000    NaN     30.000000     NaN        NaN
max    140699.000000    NaN     59.000000     NaN        NaN

        Academic Pressure  Work Pressure          CGPA  Study Satisfaction  \
count        27901.000000   27901.000000  27901.000000        27901.000000
unique                NaN            NaN           NaN                 NaN
top                   NaN            NaN           NaN                 NaN
freq                  NaN            NaN           NaN                 NaN
mean             3.141214       0.000430      7.656104            2.943837
std              1.381465       0.043992      1.470707            1.361148
min              0.000000       0.000000      0.000000            0.000000
25%              2.000000       0.000000      6.290000            2.000000
50%              3.000000       0.000000      7.770000            3.000000
75%              4.000000       0.000000      8.920000            4.000000
max              5.000000       5.000000     10.000000            5.000000

        Job Satisfaction       Sleep Duration Dietary Habits      Degree  \
count       27901.000000                27901          27901       27901
unique               NaN                    5              4          28
top                  NaN  'Less than 5 hours'      Unhealthy  'Class 12'
freq                 NaN                 8310          10317        6080
mean            0.000681                  NaN            NaN         NaN
std             0.044394                  NaN            NaN         NaN
min             0.000000                  NaN            NaN         NaN
25%             0.000000                  NaN            NaN         NaN
50%             0.000000                  NaN            NaN         NaN
75%             0.000000                  NaN            NaN         NaN
max             4.000000                  NaN            NaN         NaN

       Have you ever had suicidal thoughts ?  Work/Study Hours  \
count                                  27901      27901.000000
unique                                     2               NaN
top                                      Yes               NaN
freq                                   17656               NaN
mean                                     NaN          7.156984
std                                      NaN          3.707642
min                                      NaN          0.000000
25%                                      NaN          4.000000
50%                                      NaN          8.000000
75%                                      NaN         10.000000
```

```
max                                      NaN        12.000000

       Financial Stress Family History of Mental Illness    Depression
count             27901                            27901  27901.000000
unique                6                                2           NaN
top                 5.0                               No           NaN
freq               6715                            14398           NaN
mean                NaN                              NaN      0.585499
std                 NaN                              NaN      0.492645
min                 NaN                              NaN      0.000000
25%                 NaN                              NaN      0.000000
50%                 NaN                              NaN      1.000000
75%                 NaN                              NaN      1.000000
max                 NaN                              NaN      1.000000

Missing values per column:
 id                                      0
Gender                                  0
Age                                     0
City                                    0
Profession                              0
Academic Pressure                       0
Work Pressure                           0
CGPA                                    0
Study Satisfaction                      0
Job Satisfaction                        0
Sleep Duration                          0
Dietary Habits                          0
Degree                                  0
Have you ever had suicidal thoughts ?   0
Work/Study Hours                        0
Financial Stress                        0
Family History of Mental Illness        0
Depression                              0
dtype: int64
```

[ ]:

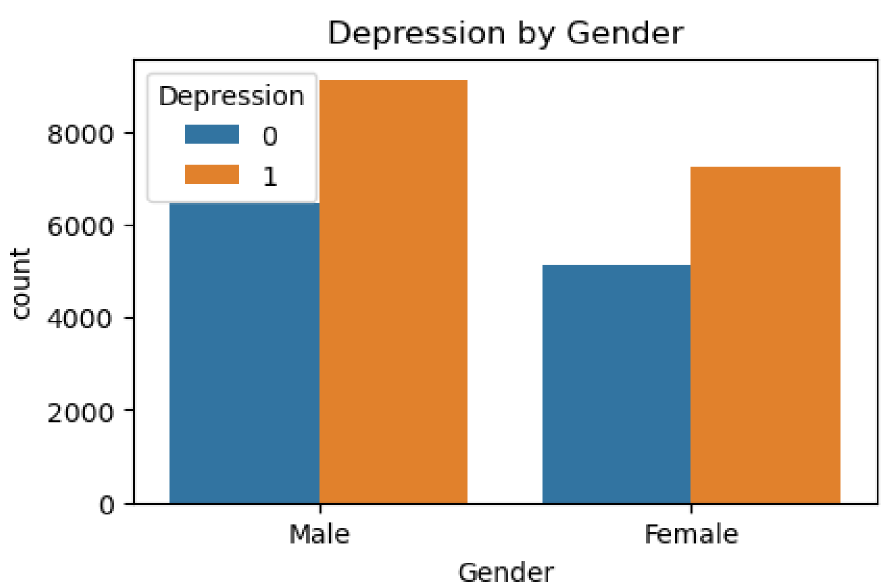
```
#  Target/Features Split & Basic Insights

target_col = 'Depression'
target = df[target_col]
features = df.drop([target_col], axis=1)
num_cols = features.select_dtypes(include=[np.number]).columns.tolist()

print("\n===== BASIC INSIGHTS =====")
print(f"Number of students: {df.shape[0]}")
print(f"Number of features: {df.shape[1]}")
print(f"Unique values in target column: {df[target_col].unique()}")
```

```
print("\nTarget value counts:\n", target.value_counts())
```

```
===== BASIC INSIGHTS =====
Number of students: 27901
Number of features: 18
Unique values in target column: [1 0]

Target value counts:
 Depression
1    16336
0    11565
Name: count, dtype: int64
```

[5]:
```
# Gender and other categorical breakdowns if present
if 'Gender' in df.columns:
    print("\n===== Gender Breakdown =====\n", df['Gender'].value_counts())
    plt.figure(figsize=(5,3))
    sns.countplot(x='Gender', data=df, hue=target_col)
    plt.title('Depression by Gender')
    plt.show()
```

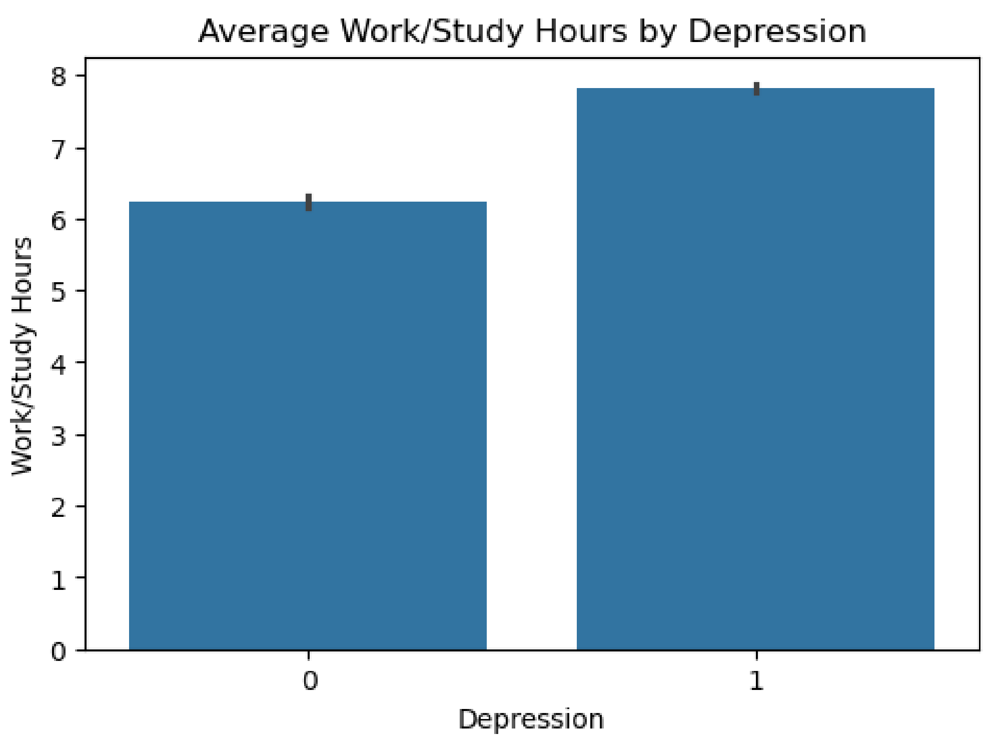
```
===== Gender Breakdown =====
 Gender
Male      15547
Female    12354
Name: count, dtype: int64
```



[6]:
```python
# Group means for key variables
print("\n===== Mean values by Depression =====\n")
print(df.groupby(target_col).mean(numeric_only=True))
```

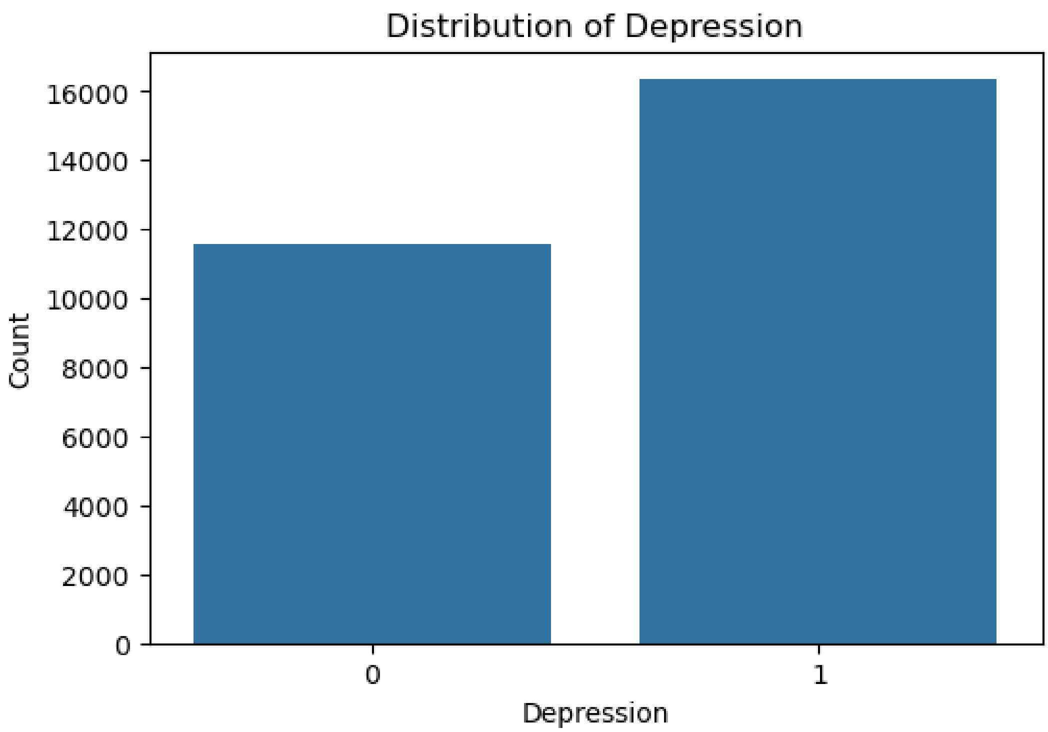
```
===== Mean values by Depression =====

                      id        Age  Academic Pressure  Work Pressure  \
Depression
0           70397.561089  27.142412           2.361608       0.000605
1           70473.715536  24.887733           3.693132       0.000306

                CGPA  Study Satisfaction  Job Satisfaction  Work/Study Hours
Depression
0           7.617282            3.215564          0.000865          6.237959
1           7.683588            2.751469          0.000551          7.807603
```

[7]:
```python
# Bar plot of average study hours (if present) by depression
if 'Work/Study Hours' in df.columns:
    plt.figure(figsize=(6,4))
    sns.barplot(x=target_col, y='Work/Study Hours', data=df)
    plt.title('Average Work/Study Hours by Depression')
    plt.show()
```

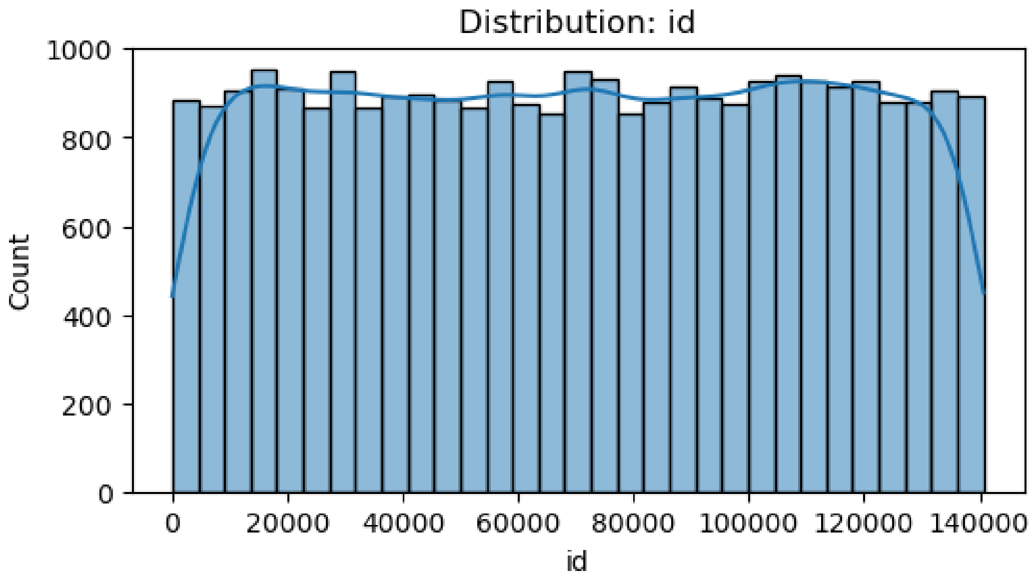


1 EDA Visualizations

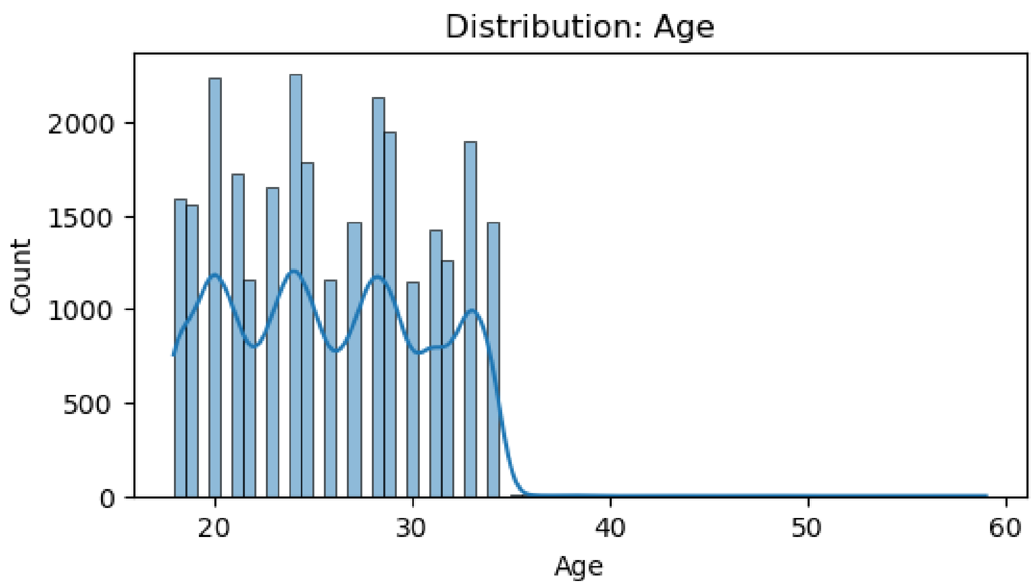
[8]:
```
# Visualize target distribution
plt.figure(figsize=(6, 4))
sns.countplot(x=target_col, data=df)
plt.title('Distribution of Depression')
plt.xlabel('Depression')
plt.ylabel('Count')
plt.show()
```



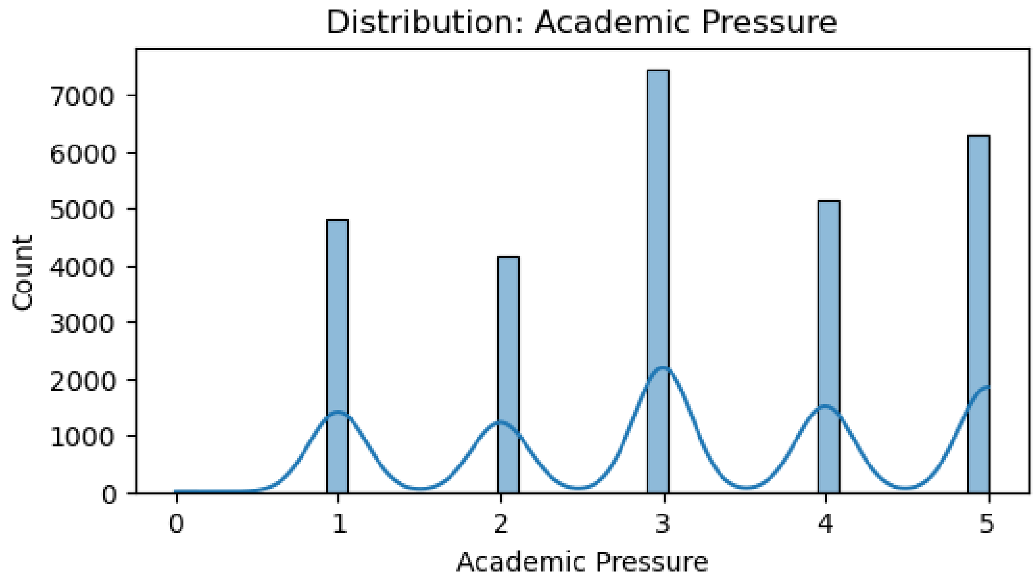
[9]:
```
# Histograms for numerical features
for col in num_cols:
    plt.figure(figsize=(6, 3))
    sns.histplot(df[col], kde=True)
    plt.title(f'Distribution: {col}')
    plt.show()
```

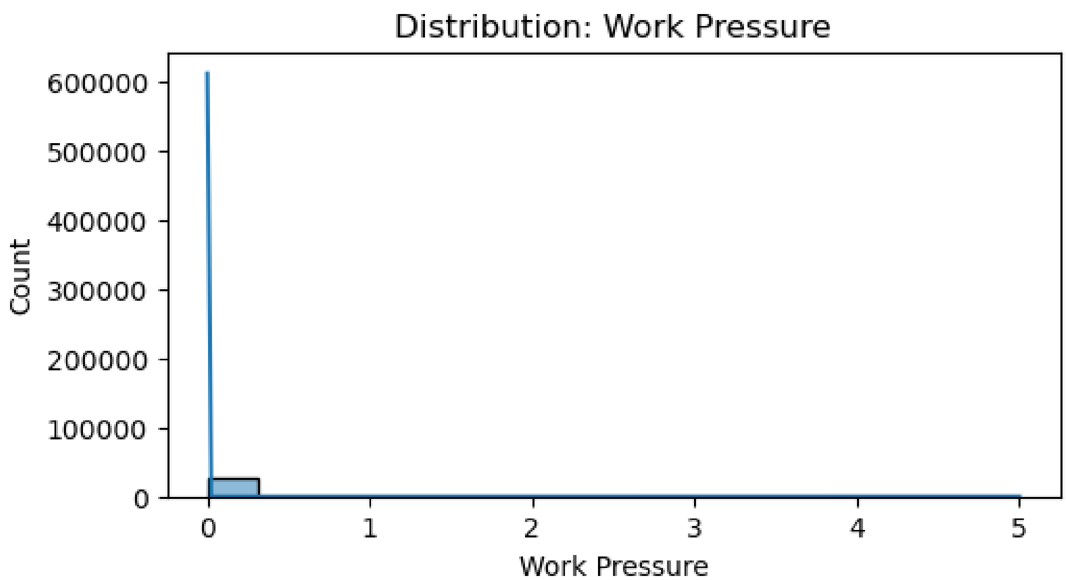


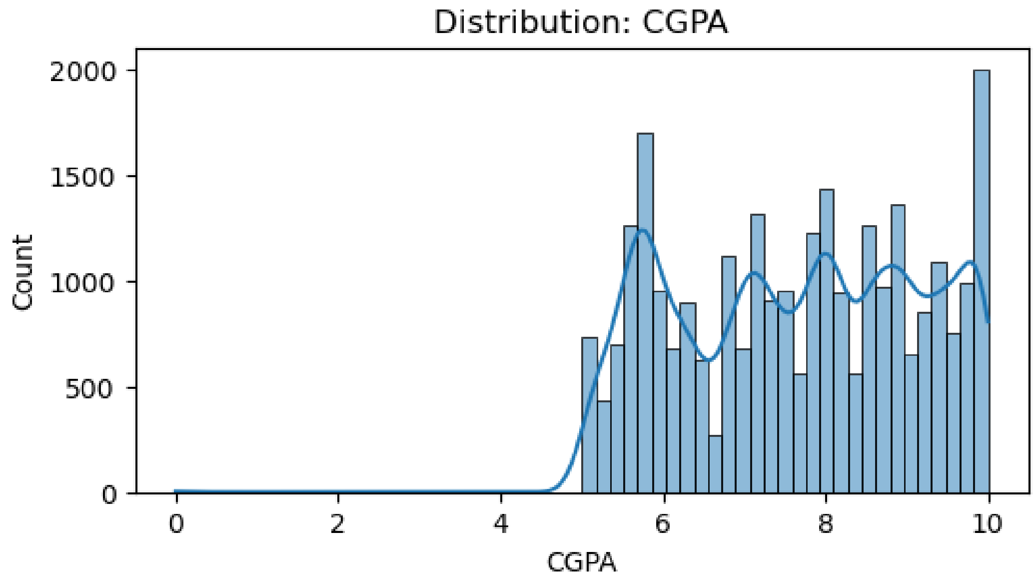
Distribution: id
1000
800
600
400
200
0
Count
0
20000
40000
60000
80000
100000
120000
140000
id



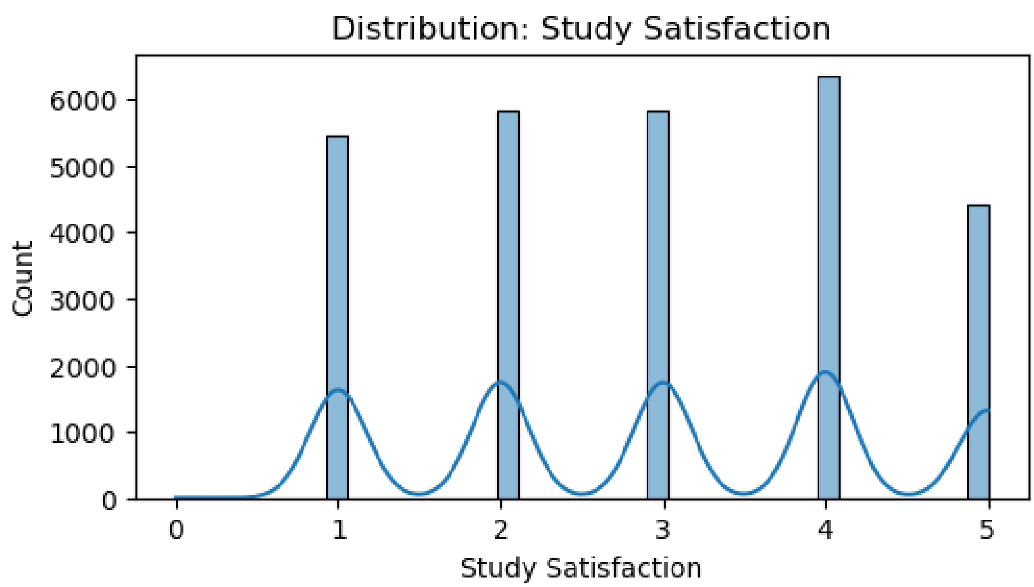
Distribution: Age
2000
1500
1000
500
0
Count
20
30
40
50
60
Age


Distribution: Academic Pressure
Count
7000
6000
5000
4000
3000
2000
1000
0
0
1
2
3
4
5
Academic Pressure

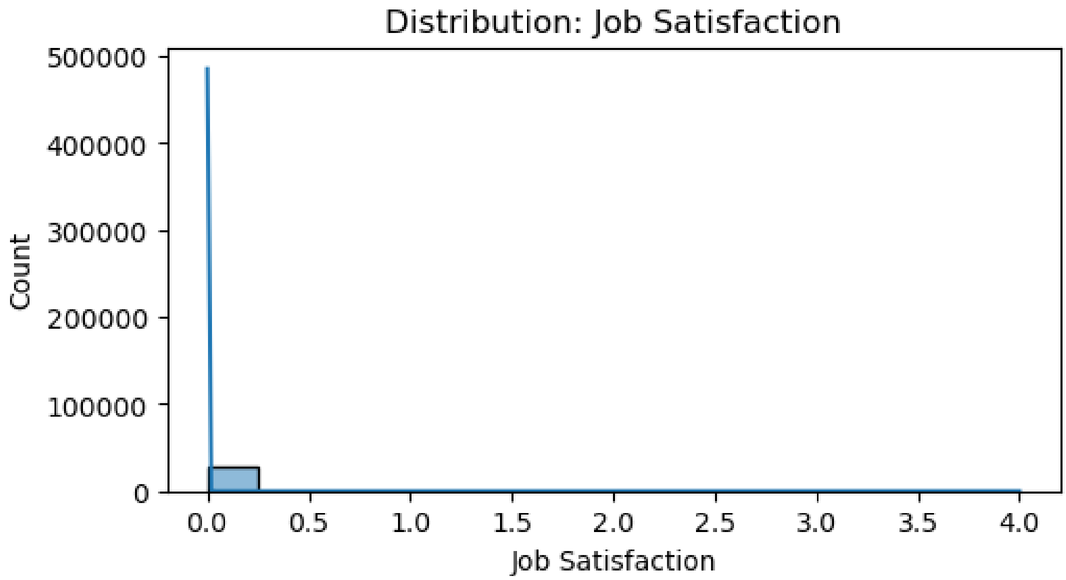

Distribution: Work Pressure
Count
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
5
Work Pressure

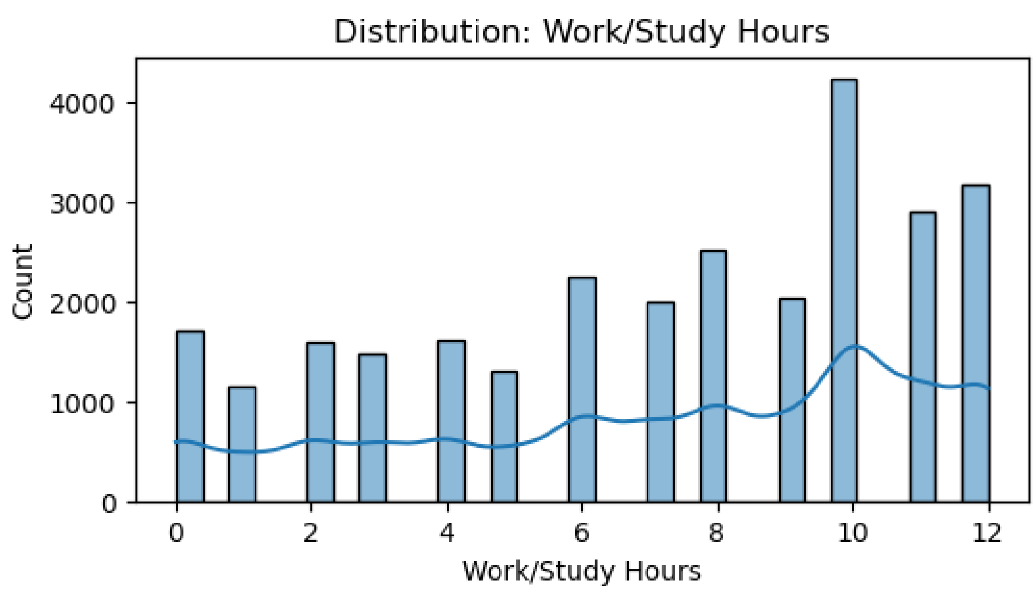


Distribution: CGPA
2000
1500
1000
500
0
Count
0
2
4
6
8
10
CGPA

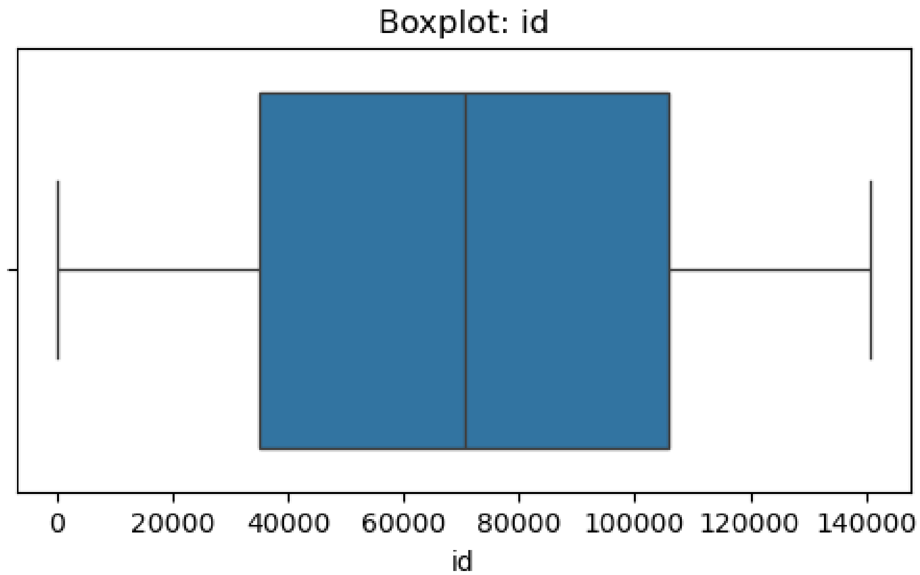


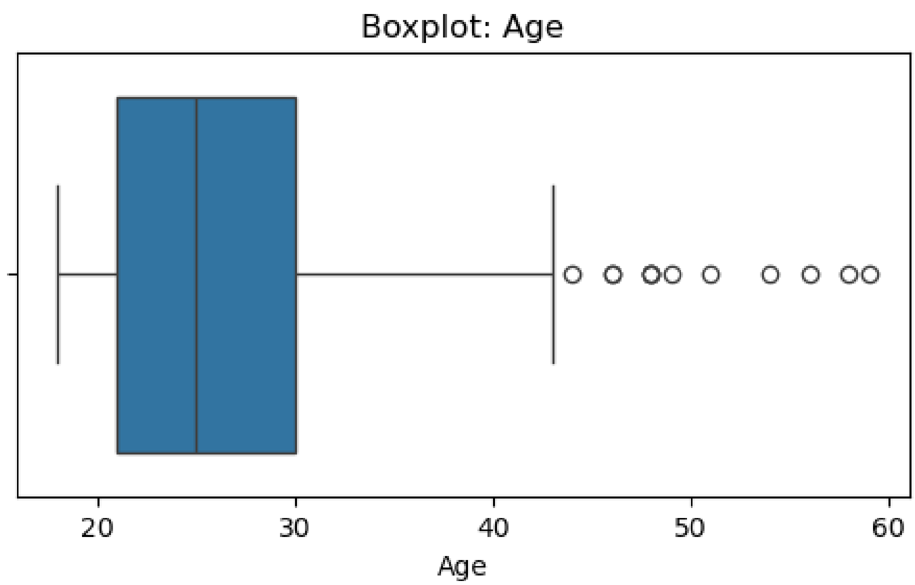
Distribution: Study Satisfaction
6000
5000
4000
3000
2000
1000
0
Count
0
1
2
3
4
5
Study Satisfaction

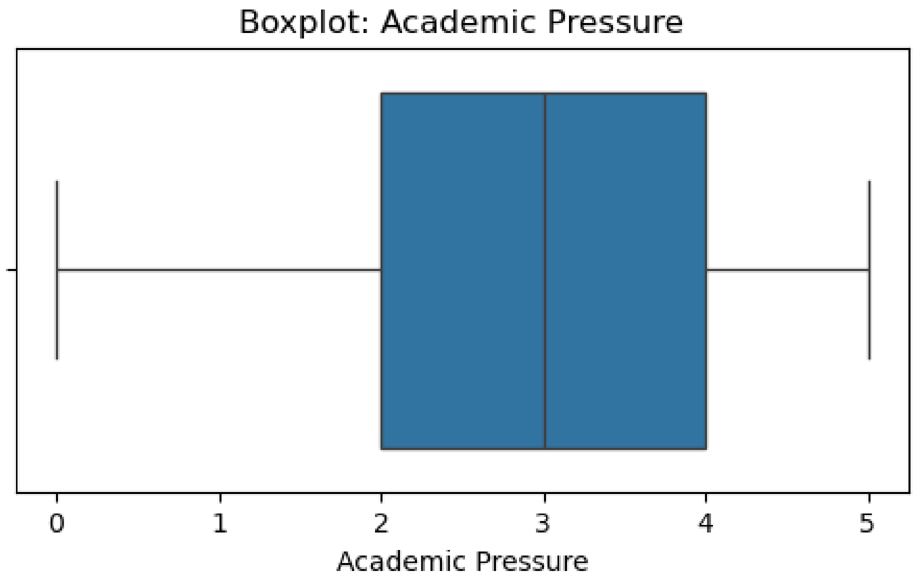


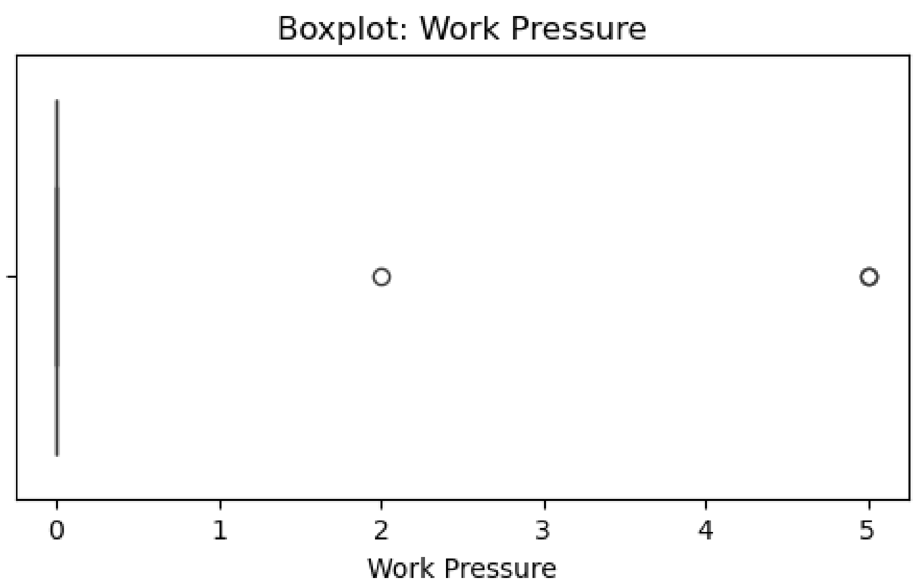


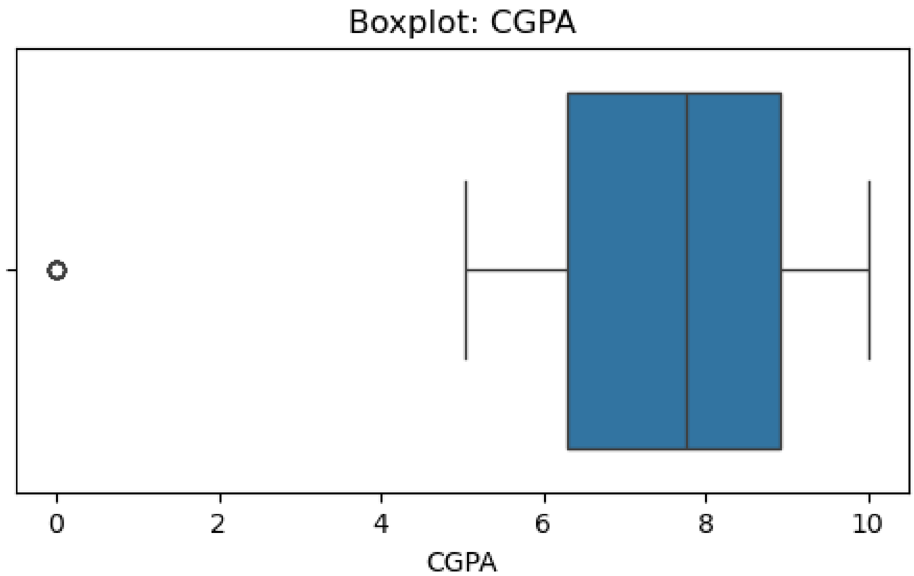
[10]:
```
# Boxplots for outlier inspection
for col in num_cols:
    plt.figure(figsize=(6, 3))
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot: {col}')
    plt.show()
```

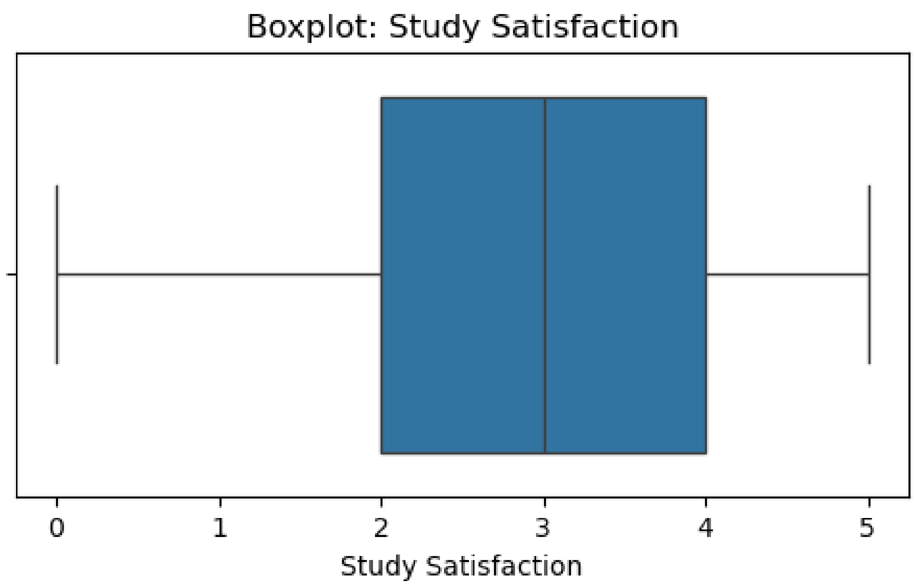

Boxplot: id
0
20000
40000
60000
80000
100000
120000
140000
id

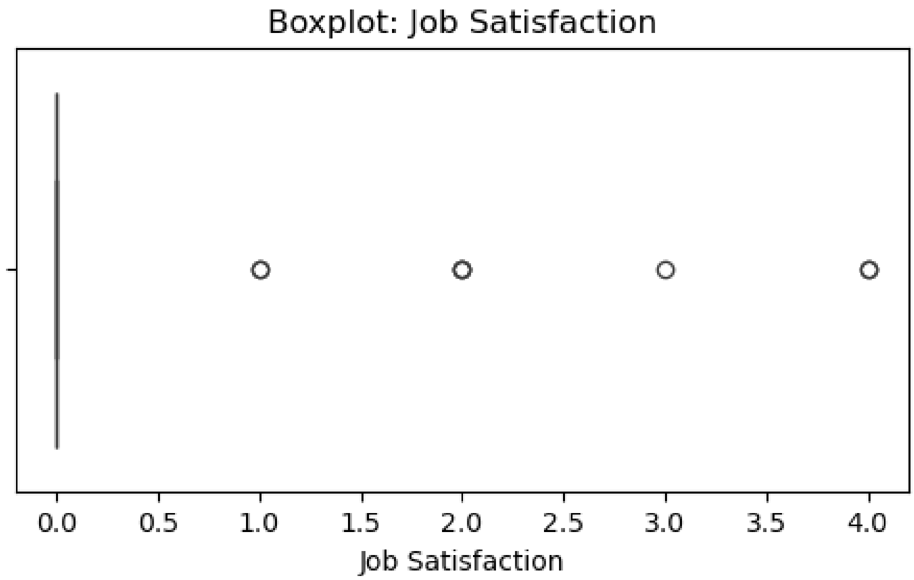

Boxplot: Age
20
30
40
50
60
Age

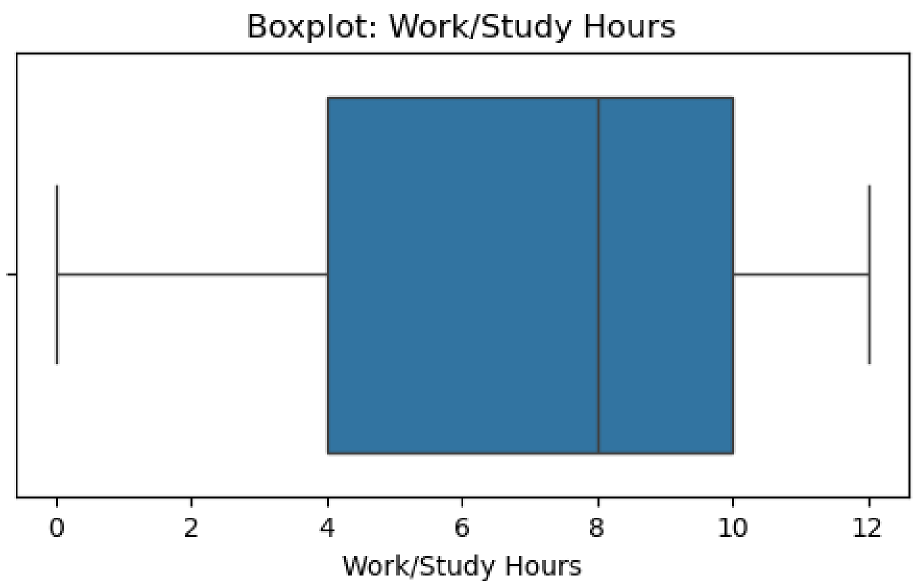

Boxplot: Academic Pressure
0
1
2
3
4
5
Academic Pressure


Boxplot: Work Pressure
0
1
2
3
4
5
Work Pressure

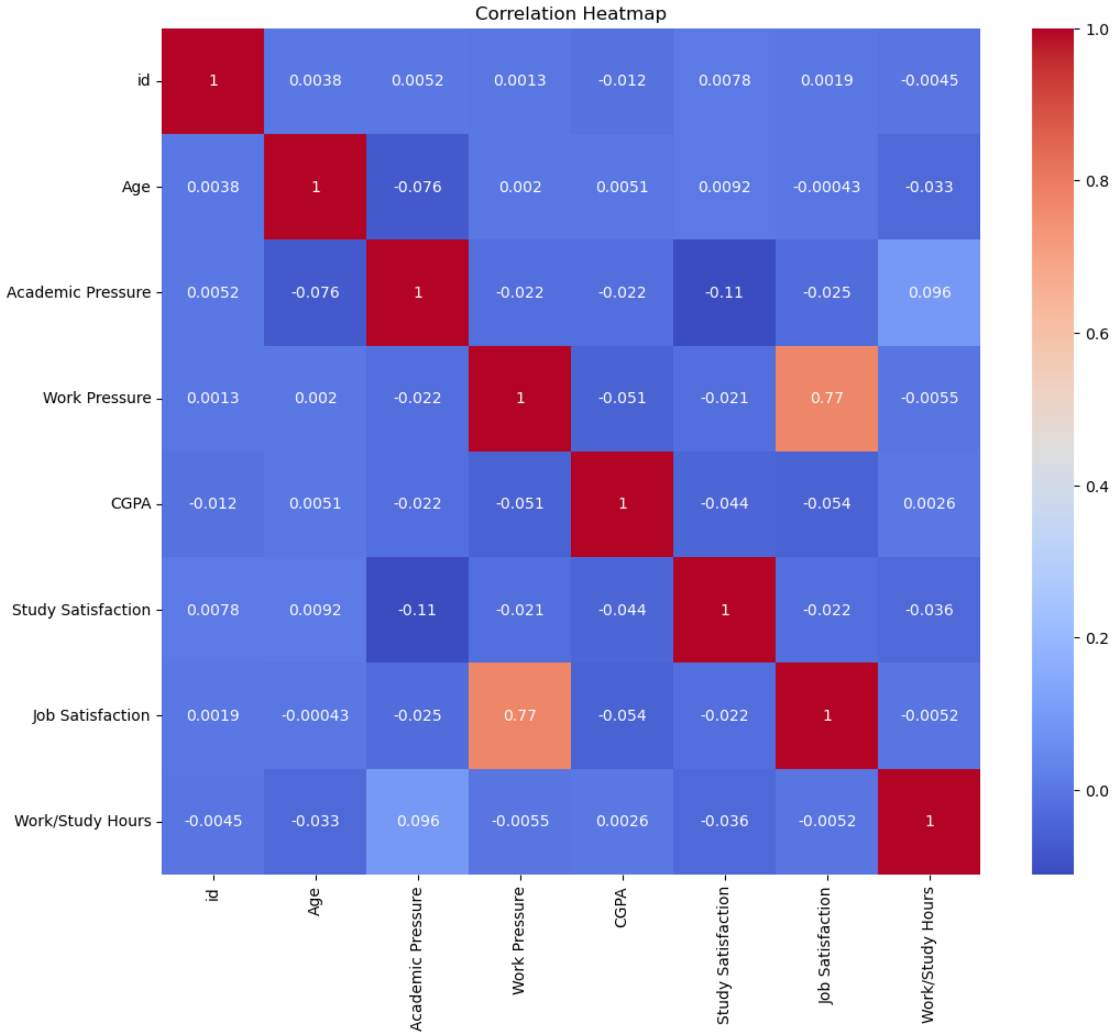

Boxplot: CGPA
0
2
4
6
8
10
CGPA


Boxplot: Study Satisfaction
0
1
2
3
4
5
Study Satisfaction





[11]:
```
# Correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df[num_cols].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



2 Data Cleaning & Preprocessing

[ ]:
```
# Fills missing numerics with median (should be none)
features[num_cols] = features[num_cols].fillna(features[num_cols].median())

# Fills missing categoricals with mode (should be none)
cat_cols = features.select_dtypes(include=['object']).columns
features[cat_cols] = features[cat_cols].fillna(features[cat_cols].mode().
    ↪iloc[0])

# Encode categorical variables
le = LabelEncoder()
for col in cat_cols:
    features[col] = le.fit_transform(features[col].astype(str))
```

```
print("\nAfter encoding, features shape:", features.shape)
print("Sample after encoding:\n", features.head())
```

```
After encoding, features shape: (27901, 17)
Sample after encoding:
    id  Gender   Age  City  Profession  Academic Pressure  Work Pressure  CGPA  \
0   2       1  33.0    51          12                5.0            0.0  8.97
1   8       0  24.0     5          12                2.0            0.0  5.90
2  26       1  31.0    44          12                3.0            0.0  7.03
3  30       0  28.0    49          12                3.0            0.0  5.59
4  32       0  25.0    18          12                4.0            0.0  8.13

   Study Satisfaction  Job Satisfaction  Sleep Duration  Dietary Habits  \
0                 2.0               0.0               0               0
1                 5.0               0.0               0               1
2                 5.0               0.0               2               0
3                 2.0               0.0               1               1
4                 3.0               0.0               0               1

   Degree  Have you ever had suicidal thoughts ?  Work/Study Hours  \
0       4                                      1               3.0
1      11                                      0               3.0
2       6                                      0               9.0
3       8                                      1               4.0
4      17                                      1               1.0

   Financial Stress  Family History of Mental Illness
0                 0                                 0
1                 1                                 1
2                 0                                 1
3                 4                                 1
4                 0                                 0
```

[13]:
```
#  Train/Test Split and Scaling

scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

X_train, X_test, y_train, y_test = train_test_split(
    features_scaled, target, test_size=0.2, random_state=42, stratify=target
)
print("Train class balance:\n", y_train.value_counts())
print("Test class balance:\n", y_test.value_counts())
```

```
Train class balance:
```

```
 Depression
1    13068
0     9252
Name: count, dtype: int64
Test class balance:
 Depression
1    3268
0    2313
Name: count, dtype: int64
```

3 Modeling & Evaluation

[14]:
```python
# Logistic Regression
logreg = LogisticRegression(max_iter=1000)
logreg.fit(X_train, y_train)
y_pred_logreg = logreg.predict(X_test)
print("\nLogistic Regression Classification Report:\n",␣
 ↪classification_report(y_test, y_pred_logreg))
```

```
Logistic Regression Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.79      0.81      2313
           1       0.86      0.88      0.87      3268

    accuracy                           0.84      5581
   macro avg       0.84      0.84      0.84      5581
weighted avg       0.84      0.84      0.84      5581
```

[15]:
```python
# Decision Tree with GridSearchCV
dt_param_grid = {'max_depth': [None, 5, 10, 15], 'min_samples_split': [2, 5,␣
 ↪10]}
dt_grid = GridSearchCV(DecisionTreeClassifier(random_state=42), dt_param_grid,␣
 ↪cv=5)
dt_grid.fit(X_train, y_train)
y_pred_dt = dt_grid.predict(X_test)
print("\nDecision Tree Best Params:", dt_grid.best_params_)
print("Decision Tree Classification Report:\n", classification_report(y_test,␣
 ↪y_pred_dt))
```

```
Decision Tree Best Params: {'max_depth': 5, 'min_samples_split': 2}
Decision Tree Classification Report:
               precision    recall  f1-score   support

           0       0.80      0.76      0.78      2313
```

```
           1       0.84      0.86      0.85      3268

    accuracy                           0.82      5581
   macro avg       0.82      0.81      0.81      5581
weighted avg       0.82      0.82      0.82      5581
```

[16]:
```
# Random Forest with GridSearchCV
rf_param_grid = {'n_estimators': [50, 100, 200], 'max_depth': [None, 10, 20],
 ↪'min_samples_split': [2, 5, 10]}
rf_grid = GridSearchCV(RandomForestClassifier(random_state=42), rf_param_grid,
 ↪cv=5)
rf_grid.fit(X_train, y_train)
y_pred_rf = rf_grid.predict(X_test)
print("\nRandom Forest Best Params:", rf_grid.best_params_)
print("Random Forest Classification Report:\n", classification_report(y_test,
 ↪y_pred_rf))
```

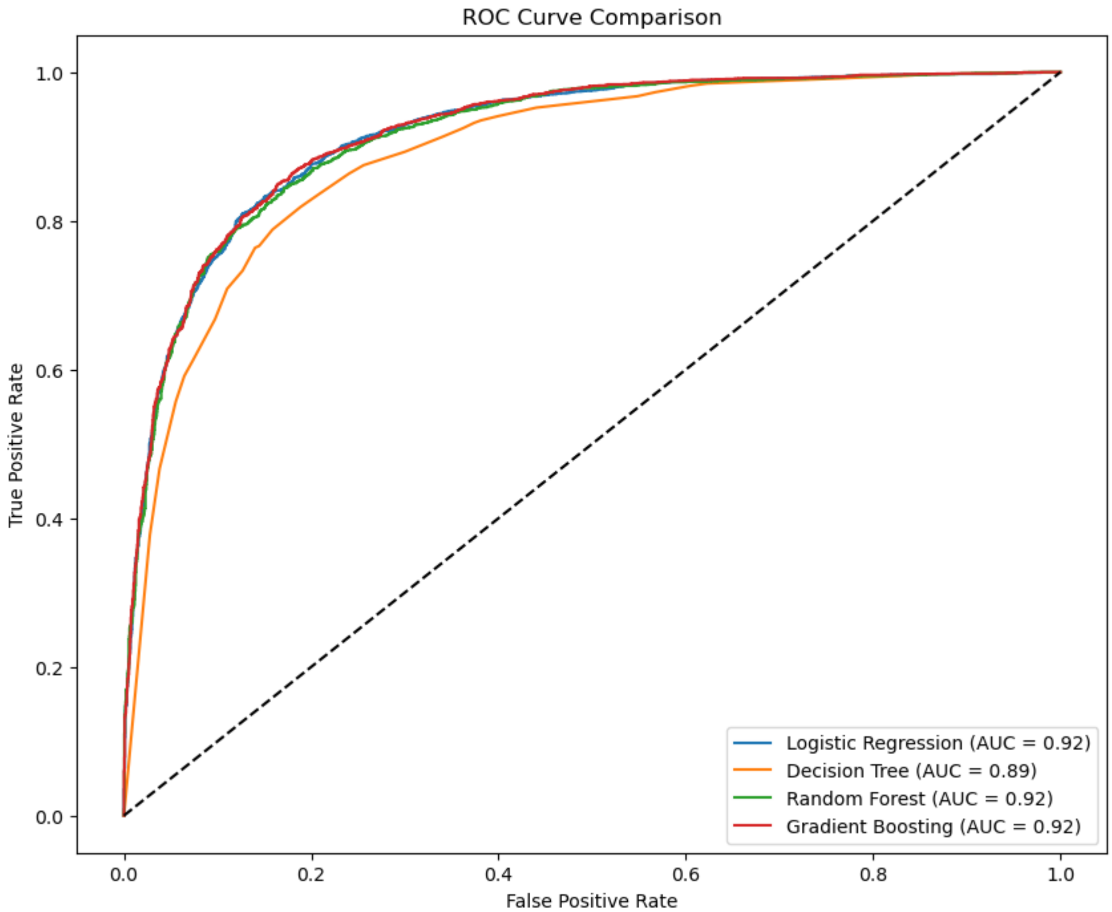
```
Random Forest Best Params: {'max_depth': 20, 'min_samples_split': 10,
'n_estimators': 200}
Random Forest Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.78      0.80      2313
           1       0.85      0.88      0.86      3268

    accuracy                           0.84      5581
   macro avg       0.84      0.83      0.83      5581
weighted avg       0.84      0.84      0.84      5581
```

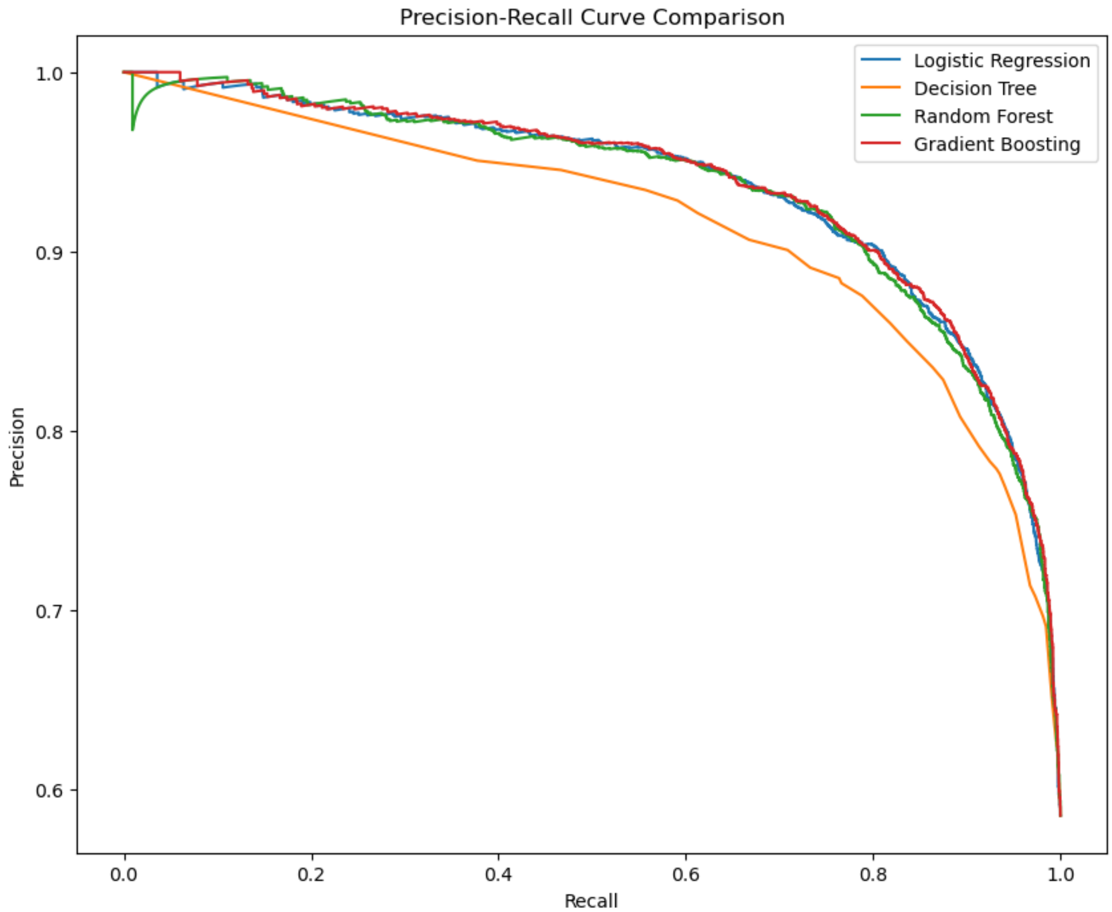
[17]:
```
# Gradient Boosting
gb = GradientBoostingClassifier(random_state=42)
gb.fit(X_train, y_train)
y_pred_gb = gb.predict(X_test)
print("\nGradient Boosting Classification Report:\n",
 ↪classification_report(y_test, y_pred_gb))
```

```
Gradient Boosting Classification Report:
               precision    recall  f1-score   support

           0       0.83      0.79      0.81      2313
           1       0.86      0.88      0.87      3268

    accuracy                           0.85      5581
   macro avg       0.84      0.84      0.84      5581
```

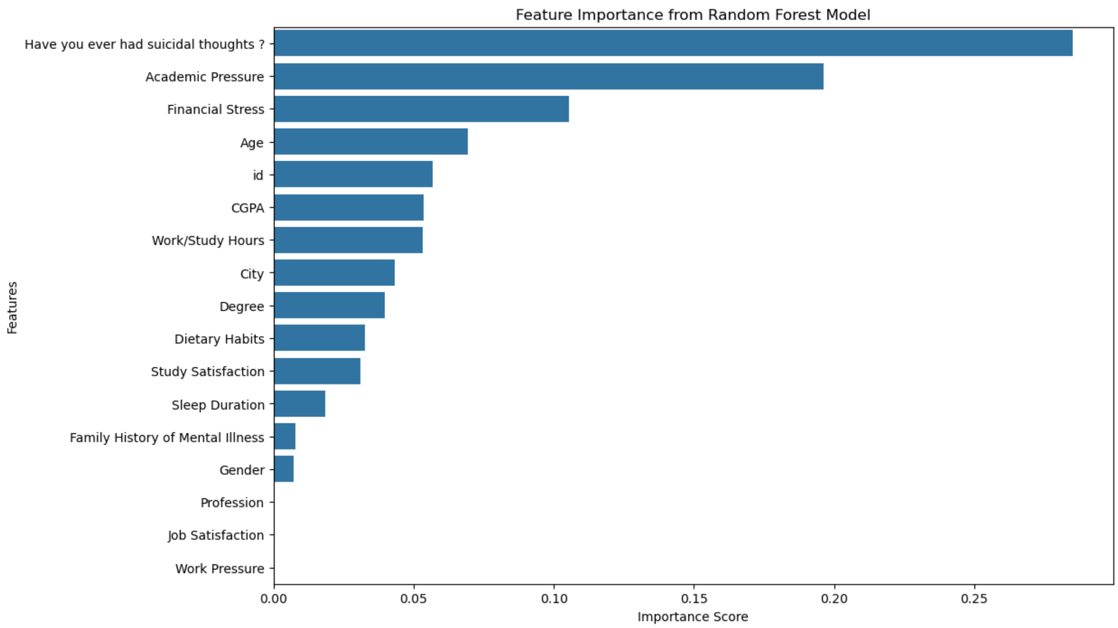
```
weighted avg       0.85      0.85      0.85      5581
```

[18]:
```
#  ROC and Precision-Recall Curves

plt.figure(figsize=(10, 8))
for model, name in zip([logreg, dt_grid, rf_grid, gb],
                       ['Logistic Regression', 'Decision Tree', 'Random␣
 ↪Forest', 'Gradient Boosting']):
    y_pred_prob = model.predict_proba(X_test)[:, 1]
    fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
    auc = roc_auc_score(y_test, y_pred_prob)
    plt.plot(fpr, tpr, label=f'{name} (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate'); plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison')
plt.legend(loc='lower right')
plt.show()

plt.figure(figsize=(10, 8))
for model, name in zip([logreg, dt_grid, rf_grid, gb],
                       ['Logistic Regression', 'Decision Tree', 'Random␣
 ↪Forest', 'Gradient Boosting']):
    y_pred_prob = model.predict_proba(X_test)[:, 1]
    precision, recall, _ = precision_recall_curve(y_test, y_pred_prob)
    plt.plot(recall, precision, label=f'{name}')
plt.xlabel('Recall'); plt.ylabel('Precision')
plt.title('Precision-Recall Curve Comparison')
plt.legend(loc='best')
plt.show()
```


ROC Curve Comparison
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression (AUC = 0.92)
Decision Tree (AUC = 0.89)
Random Forest (AUC = 0.92)
Gradient Boosting (AUC = 0.92)

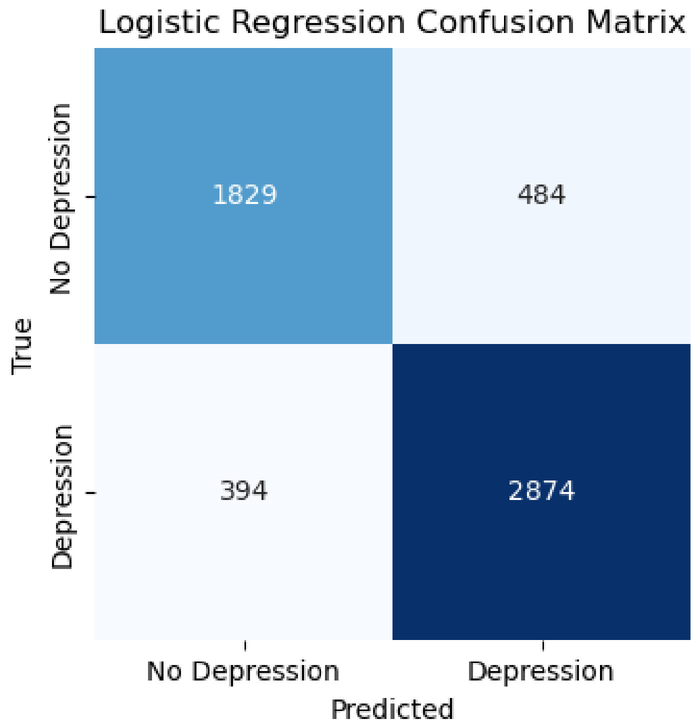


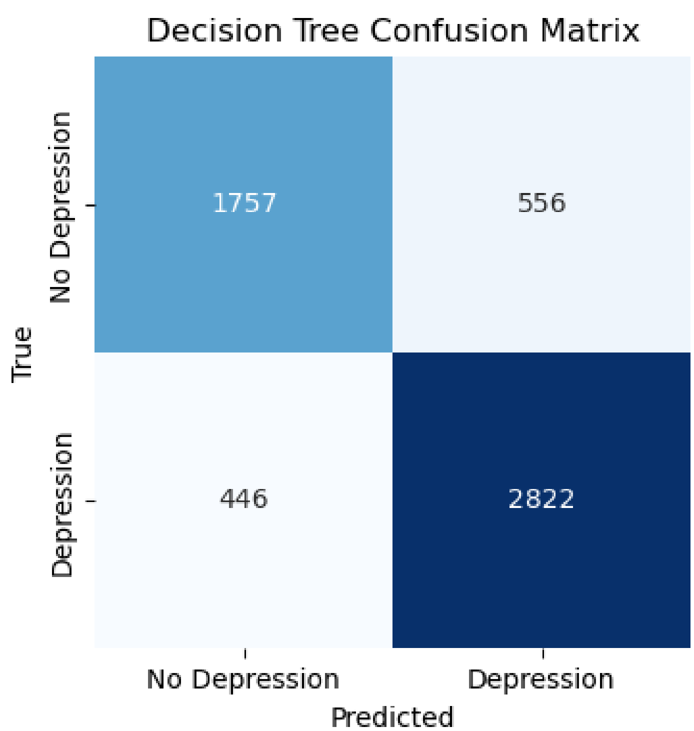
[19]:
```
# 9. Feature Importance (Random Forest)

importance = rf_grid.best_estimator_.feature_importances_
indices = np.argsort(importance)[::-1]
plt.figure(figsize=(12, 8))
sns.barplot(x=importance[indices], y=features.columns[indices])
plt.title('Feature Importance from Random Forest Model')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.show()
```

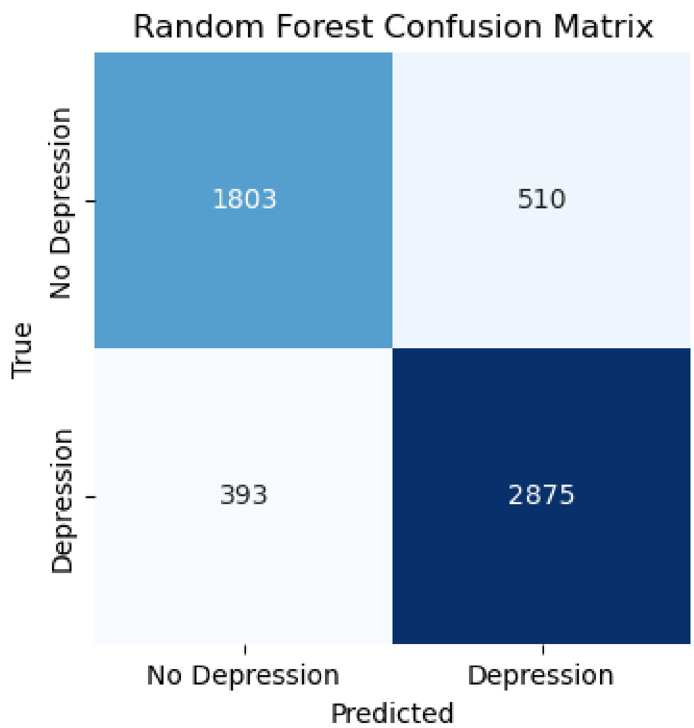


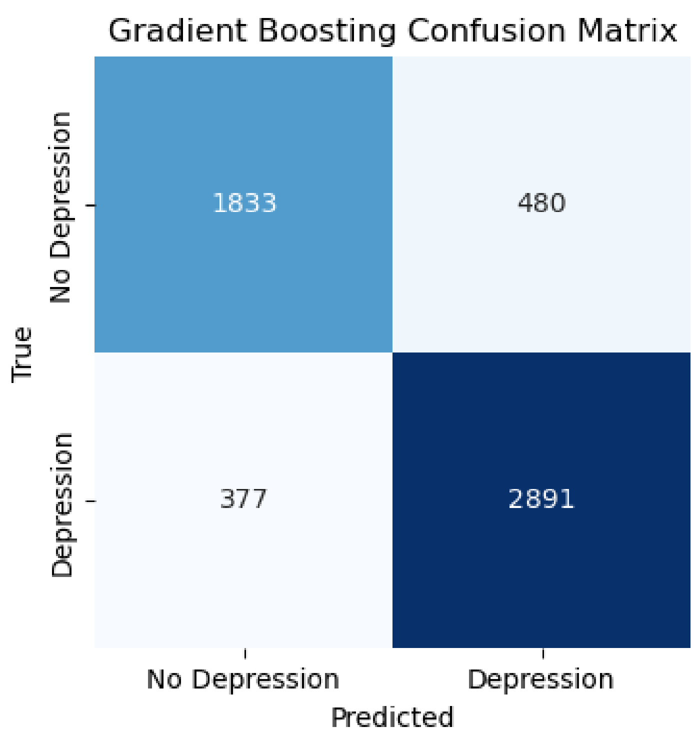
[20]:

```python
# 10. Confusion Matrices

def plot_cm(cm, classes, title):
    plt.figure(figsize=(4,4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
                xticklabels=classes, yticklabels=classes)
    plt.title(title)
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

models = [logreg, dt_grid, rf_grid, gb]
model_names = ['Logistic Regression', 'Decision Tree', 'Random Forest',␣
 ↪'Gradient Boosting']
preds = [y_pred_logreg, y_pred_dt, y_pred_rf, y_pred_gb]

for name, model, pred in zip(model_names, models, preds):
    cm = confusion_matrix(y_test, pred)
    plot_cm(cm, classes=['No Depression', 'Depression'], title=f'{name}␣
 ↪Confusion Matrix')
```


Logistic Regression Confusion Matrix
True
No Depression
Depression
1829
484
394
2874
No Depression
Depression
Predicted


Decision Tree Confusion Matrix
True
No Depression
Depression
1757
556
446
2822
No Depression
Depression
Predicted


Random Forest Confusion Matrix
True
No Depression
Depression
1803
510
393
2875
No Depression
Depression
Predicted


Gradient Boosting Confusion Matrix
True
No Depression
Depression
1833
480
377
2891
No Depression
Depression
Predicted

[21]:
```python
# 11. Accuracy Summary

print("\n===== FINAL MODEL COMPARISON (ACCURACY) =====")
for name, pred in zip(model_names, preds):
    print(f"{name}: {accuracy_score(y_test, pred):.3f}")
```

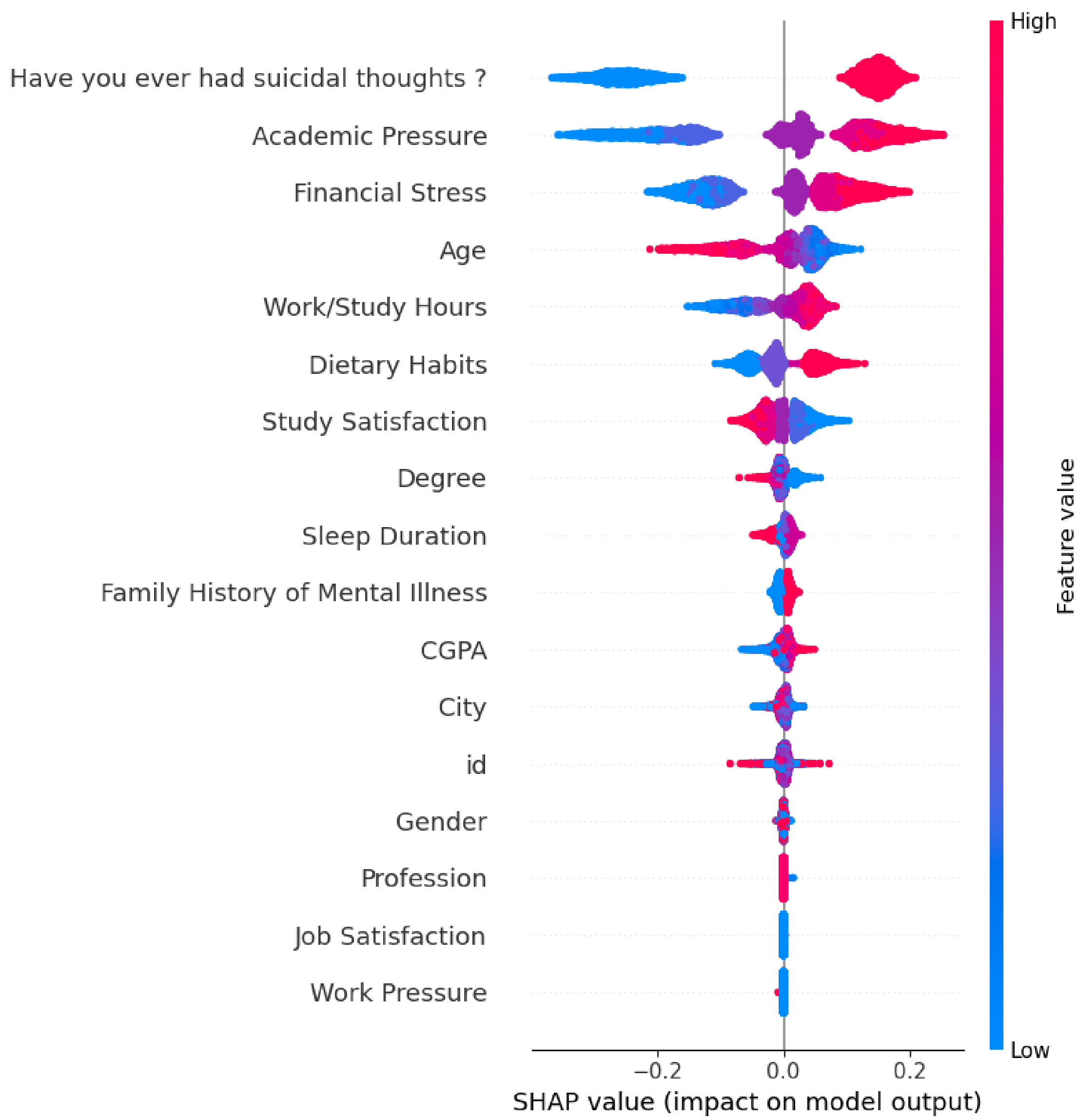
```
===== FINAL MODEL COMPARISON (ACCURACY) =====
Logistic Regression: 0.843
Decision Tree: 0.820
Random Forest: 0.838
Gradient Boosting: 0.846
```

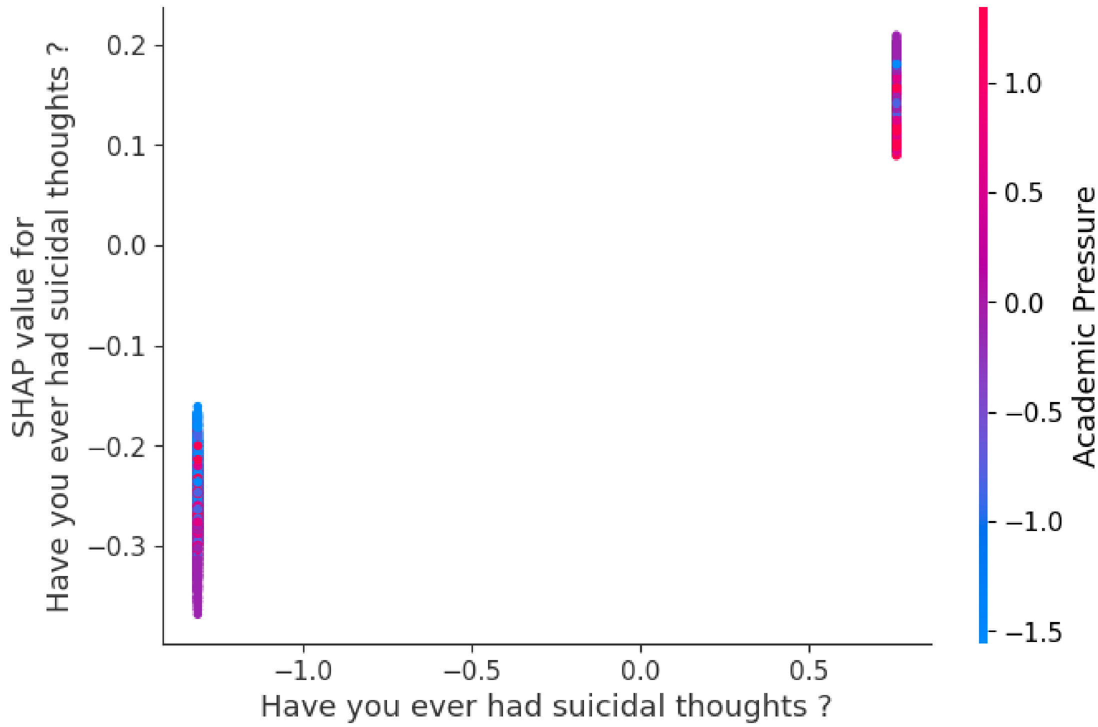
[ ]:
```python
# Reconstructing the dataframe
X_train, X_test, y_train, y_test = train_test_split(
    features, target, test_size=0.2, random_state=42, stratify=target
)

# Standardize (made sure to keep as DataFrame)
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = pd.DataFrame(scaler.fit_transform(X_train), columns=features.
 ↪columns, index=X_train.index)
X_test_scaled = pd.DataFrame(scaler.transform(X_test), columns=features.
 ↪columns, index=X_test.index)

# Ensuring X_test_scaled is a DataFrame
# shap_values is a list of two arrays: one for each class
print(type(shap_values))
print(len(shap_values))
print(shap_values[0].shape)
print(shap_values[1].shape)
```

```
<class 'numpy.ndarray'>
5581
(17, 2)
(17, 2)
```

[31]:
```python
# 12. SHAP Explainability (Random Forest)
import shap

explainer = shap.TreeExplainer(rf_grid.best_estimator_)
shap_values = explainer.shap_values(X_test_scaled)

# If 3D array, take positive class only
if len(shap_values.shape) == 3 and shap_values.shape[2] == 2:
    shap_values_pos = shap_values[:, :, 1]
```

```
else:
    shap_values_pos = shap_values

shap.summary_plot(shap_values_pos, X_test_scaled, feature_names=features.
 ↪columns)

top_feature = features.columns[np.argmax(rf_grid.best_estimator_.
 ↪feature_importances_)]
shap.dependence_plot(top_feature, shap_values_pos, X_test_scaled,␣
 ↪feature_names=features.columns)
```





4 Statistical Tests

[34]:
```
from scipy.stats import ttest_ind, chi2_contingency
```

[35]:
```
# T-test: compare mean 'Work/Study Hours' for Depressed vs Not
if 'Work/Study Hours' in df.columns:
    group0 = df[df['Depression'] == 0]['Work/Study Hours']
    group1 = df[df['Depression'] == 1]['Work/Study Hours']
    t_stat, p_val = ttest_ind(group0, group1, nan_policy='omit')
    print(f"\nT-test for Work/Study Hours (Depressed vs Not): t={t_stat:.3f},␣
↪p={p_val:.4f}")
    if p_val < 0.05:
        print('Result: Significant difference.')
    else:
        print('Result: No significant difference.')
```

```
T-test for Work/Study Hours (Depressed vs Not): t=-35.620, p=0.0000
Result: Significant difference.
```

[36]:
```
# Chi-square: Gender vs Depression
if 'Gender' in df.columns:
    contingency = pd.crosstab(df['Gender'], df['Depression'])
```

```
    chi2, p, _, _ = chi2_contingency(contingency)
    print(f"\nChi-square for Gender vs Depression: chi2={chi2:.2f}, p={p:.4f}")
    if p < 0.05:
        print('Result: Significant association.')
    else:
        print('Result: No significant association.')
```

```
Chi-square for Gender vs Depression: chi2=0.08, p=0.7737
Result: No significant association.
```

5 Model Export

[37]:
```
# 14. Automated EDA Profiling Report (HTML)

# pip install ydata-profiling
from ydata_profiling import ProfileReport

profile = ProfileReport(df, title='Student Depression Dataset Profiling␣
 ↪Report', explorative=True)
profile.to_file('student_depression_eda_report.html')
print('Automated EDA report saved as student_depression_eda_report.html')
```

```
<IPython.core.display.HTML object>

Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

100%|      | 18/18 [00:00<00:00, 76.86it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]

Automated EDA report saved as student_depression_eda_report.html
```

[38]:
```
import joblib
joblib.dump(rf_grid.best_estimator_, 'student_depression_rf_model.joblib')
print('Random Forest model saved as student_depression_rf_model.joblib')
```

```
Random Forest model saved as student_depression_rf_model.joblib
```

[ ]:
```
# 16. Data Dictionary & Group Summary

# Converts known numeric-looking columns to float
for col in ['Financial Stress']:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col], errors='coerce')

# Numeric summary by Depression group
```

```python
numeric_cols = df.select_dtypes(include=[np.number]).columns
group_summary = df.groupby(target_col)[numeric_cols].agg(['mean', 'std',
 ↪'count'])
print('\nSummary stats by Depression:\n', group_summary)
group_summary.to_csv('depression_group_summary.csv')
print('Group summary saved as depression_group_summary.csv')

# Data dictionary
print('\nDATA DICTIONARY:')
for col in df.columns:
    print(f"{col}: {df[col].dtype}, unique: {df[col].nunique()}, sample:
 ↪{df[col].unique()[:3]}")

# Showing value counts for categoricals by group
cat_cols = df.select_dtypes(include='object').columns
for col in cat_cols:
    print(f"\nValue counts of '{col}' by {target_col}:")
    print(df.groupby(target_col)[col].value_counts())
```

```
Summary stats by Depression:
                      id                                Age                    \
                   mean           std  count       mean       std  count
Depression
0           70397.561089  40556.248313  11565  27.142412  4.943370  11565
1           70473.715536  40702.402805  16336  24.887733  4.658028  16336

           Academic Pressure                  Work Pressure  …  \
                        mean       std  count          mean  …
Depression                                                   …
0                   2.361608  1.252937  11565      0.000605  …
1                   3.693132  1.188834  16336      0.000306  …

           Job Satisfaction Work/Study Hours                   \
                      count             mean       std  count
Depression
0                     11565         6.237959  3.860943  11565
1                     16336         7.807603  3.450328  16336

           Financial Stress                  Depression
                       mean       std  count       mean  std  count
Depression
0                  2.518724  1.346952  11563        0.0  0.0  11565
1                  3.579553  1.333337  16335        1.0  0.0  16336

[2 rows x 30 columns]
Group summary saved as depression_group_summary.csv
```

```
DATA DICTIONARY:
id: int64, unique: 27901, sample: [ 2  8 26]
Gender: object, unique: 2, sample: ['Male' 'Female']
Age: float64, unique: 34, sample: [33. 24. 31.]
City: object, unique: 52, sample: ['Visakhapatnam' 'Bangalore' 'Srinagar']
Profession: object, unique: 14, sample: ['Student' "'Civil Engineer'"
'Architect']
Academic Pressure: float64, unique: 6, sample: [5. 2. 3.]
Work Pressure: float64, unique: 3, sample: [0. 5. 2.]
CGPA: float64, unique: 332, sample: [8.97 5.9  7.03]
Study Satisfaction: float64, unique: 6, sample: [2. 5. 3.]
Job Satisfaction: float64, unique: 5, sample: [0. 3. 4.]
Sleep Duration: object, unique: 5, sample: ["'5-6 hours'" "'Less than 5 hours'"
"'7-8 hours'"]
Dietary Habits: object, unique: 4, sample: ['Healthy' 'Moderate' 'Unhealthy']
Degree: object, unique: 28, sample: ['B.Pharm' 'BSc' 'BA']
Have you ever had suicidal thoughts ?: object, unique: 2, sample: ['Yes' 'No']
Work/Study Hours: float64, unique: 13, sample: [3. 9. 4.]
Financial Stress: float64, unique: 5, sample: [1. 2. 5.]
Family History of Mental Illness: object, unique: 2, sample: ['No' 'Yes']
Depression: int64, unique: 2, sample: [1 0]

Value counts of 'Gender' by Depression:
Depression  Gender
0           Male      6432
            Female    5133
1           Male      9115
            Female    7221
Name: count, dtype: int64

Value counts of 'City' by Depression:
Depression  City
0           Kalyan         636
            Srinagar       609
            Vasai-Virar    551
            Lucknow        514
            Agra           509
                          …
1           M.Com            1
            Mihir            1
            Mira             1
            Nalini           1
            Vaanya           1
Name: count, Length: 84, dtype: int64

Value counts of 'Profession' by Depression:
Depression  Profession
```

```
0           Student                     11562
            'Digital Marketer'              1
            Architect                       1
            Teacher                         1
1           Student                     16308
            Architect                       7
            Teacher                         5
            'Content Writer'                2
            'Digital Marketer'              2
            Chef                            2
            Doctor                          2
            Pharmacist                      2
            'Civil Engineer'                1
            'Educational Consultant'        1
            'UX/UI Designer'                1
            Entrepreneur                    1
            Lawyer                          1
            Manager                         1
Name: count, dtype: int64

Value counts of 'Sleep Duration' by Depression:
Depression  Sleep Duration
0           '7-8 hours'            2975
            'More than 8 hours'    2966
            'Less than 5 hours'    2949
            '5-6 hours'            2666
            Others                    9
1           'Less than 5 hours'    5361
            '7-8 hours'            4371
            '5-6 hours'            3517
            'More than 8 hours'    3078
            Others                    9
Name: count, dtype: int64

Value counts of 'Dietary Habits' by Depression:
Depression  Dietary Habits
0           Moderate         4363
            Healthy          4178
            Unhealthy        3020
            Others              4
1           Unhealthy        7297
            Moderate         5558
            Healthy          3473
            Others              8
Name: count, dtype: int64

Value counts of 'Degree' by Depression:
Depression  Degree
```

```
0           'Class 12'    1777
            B.Ed           846
            B.Com          653
            BCA            614
            B.Arch         607
            MSc            511
            M.Tech         501
            B.Tech         497
            MCA            485
            BHM            416
            M.Ed           406
            B.Pharm        382
            BSc            365
            M.Com          344
            LLB            315
            MBBS           292
            BBA            289
            BA             279
            BE             279
            MD             274
            M.Pharm        268
            MBA            259
            MA             254
            PhD            236
            LLM            223
            MHM             92
            ME              87
            Others          14
1           'Class 12'    4303
            B.Ed          1021
            B.Arch         871
            B.Com          853
            BCA            819
            MSc            679
            B.Tech         655
            MCA            559
            BSc            523
            M.Tech         521
            BHM            509
            B.Pharm        428
            M.Ed           415
            BBA            407
            MBBS           404
            M.Com          390
            LLB            356
            BE             334
            BA             321
            M.Pharm        314
```

```
            MBA               303
            MD                298
            MA                290
            PhD               286
            LLM               259
            MHM                99
            ME                 98
            Others             21
Name: count, dtype: int64

Value counts of 'Have you ever had suicidal thoughts ?' by Depression:
Depression  Have you ever had suicidal thoughts ?
0           No                                        7866
            Yes                                       3699
1           Yes                                      13957
            No                                        2379
Name: count, dtype: int64

Value counts of 'Family History of Mental Illness' by Depression:
Depression  Family History of Mental Illness
0           No                                  6335
            Yes                                 5230
1           Yes                                 8273
            No                                  8063
Name: count, dtype: int64
```

This notebook was converted with convert.ploomber.io